Semantic Segmentation of Railway Images Considering Temporal Continuity

Yuki Furitsu[1], Daisuke Deguchi[1], Yasutomo Kawanishi[1], Ichiro Ide[1], Hiroshi Murase[1], Hiroki Mukojima[2], and Nozomi Nagamine[2]

[1] Nagoya University, Nagoya, Japan
furitsuy@murase.is.i.nagoya-u.ac.jp
[2] Railway Technical Research Institute, Tokyo, Japan

Abstract. In this paper, we focus on the semantic segmentation of images taken from a camera mounted on the front end of trains for measuring and managing rail-side facilities. Improving the efficiency and perhaps automating such tasks are crucial as they are currently done manually. We aim to realize this by capturing information about the railway environment through the semantic segmentation of train front-view camera images. Specifically, assuming that the lateral movement of trains are smooth, we propose a method to use information from multiple frames to consider temporal continuity during semantic segmentation. Based on the densely estimated optical flow between sequential frames, the weighted mean of class likelihoods of corresponding pixels of the focused frame are calculated. We also construct a new dataset consisting of train front-view camera images and its annotations for semantic segmentation. The proposed method outperforms a conventional single-frame semantic segmentation model, and the use of class likelihoods for the frame combination also proved effective.

Keywords: Semantic segmentation · Railway · Optical flow.

1 Introduction

Railways are widely spread as a fast and mass transportation means, especially in Japan. Due to its nature, the impact of an accident once it occurs will be humanly, socially, and economically devastating, making the safety of railways a heavily emphasized issue. For such reasons, many rail-side facilities like railway signals, beacons for Automatic Train Stop system (ATS), and so on are installed. At the same time, some rail-side facilities like wire columns need daily maintenance to make sure they do not obstruct the trains' path. However, geological / geometrical positions of such facilities and objects are currently collected manually, which is a time-consuming and expensive task. Therefore, technological improvements in measuring the exact location of rail-side facilities and improving the efficiency of, perhaps fully automating, the maintenance of such facilities are essential.

To meet such needs, some researches have been performed to apply Mobile Mapping System (MMS) to railways [7]. Dense 3D point clouds can be obtained

from an MMS vehicle loaded on a railway bogie. Using such point clouds, it is possible to take close measurements of rail-side facilities like rail positions and station platforms. However, a specially designed and expensive equipment (MMS vehicle) is required in such approach, and also measurements cannot be taken during railway operation hours as a railway bogie must be pulled slowly. Meanwhile, since visible images cannot be taken during night time, texture information will be unavailable for maintenance tasks.

There is also an approach to combine semantic segmentation and 3D reconstruction to obtain a class-labeled 3D map [5]. However, this approach cannot be directly applied to the railway environment since the accuracy of semantic segmentation of such environment is insufficient for practical use.

To tackle this problem, we aim to improve the accuracy of semantic segmentation of the railway environment. We consider using train front-view camera images taken from a camera mounted in front of the driver's seat on a normally operated train. Such cameras can also be used for other purposes like obstacle detection, and also need little cost to introduce as it does not require large scale remodeling of trains or an expansion of ground facilities. Also, some recent trains already are equipped with driving recorders consisting of similar cameras. Combining train front-view cameras and the recent technology of semantic segmentation, a method of recognizing both the class of objects and their locations within an image, we aim to recognize the 3D space of railways including rail-side facilities. Specifically, we project the semantic segmentation result onto a 3D point cloud to obtain a class-labeled 3D map of the railway environment.

Semantic segmentation is a task of allocating labels to each pixel within an image. Many models have been develped for this purpose in recent years, with some prominent examples being the Fully Convolutional Network (FCN) [6], SegNet [1], and DeepLabv3+ [2]. Such state-of-the-art models have recorded high segmentation accuracies on the Cityscapes [3] dataset, consisting of in-vehicle camera images. In our research, we apply this technology to the railway enviroinment.

In particular, we take into account that in sequential railway images, the same object tends to appear continuously and with small movement, and thus use semantic segmentation results of not only the current frame, but frames prior to and after it to better capture the information of objects. Our proposed model enhances the "raw" semantic segmentation outputs of state-of-the-art methods by considering the temporal continuity of such sequential frames using dense optical flow.

Also, to the authors' best of knowledge, there is no dataset available for the semantic segmentation of the railway environment. As a matter of fact, some researches have been performed on recognizing materials of objects that appear around rail tracks using semantic segmentation [4]. However, such research is insufficient for understanding the railway envirionment as a whole. Therefore, we build a novel dataset consisting of train front-view camera images, and its annotations for semantic segmentation. Though the dataset size is comparatively

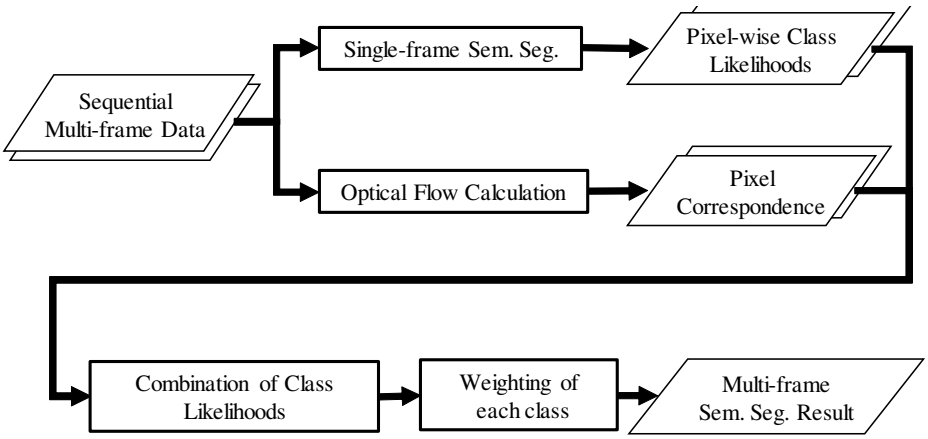

Fig. 1. Diagram showing the overview of the proposed method.

small, we show that the use of the dataset boosts the semantic segmentation accuracy of such environment.

Furthermore, we use the obtained semantic segmentation result and project it onto a 3D map. This map can be obtained by applying Structure from Motion (SfM) to train front-view camera images, and combining it with semantic labels will allow us to understand the railway environment at ease.

To summarize, the main contributions of this paper are:

1. A novel semantic segmentation method targeting train front-view camera images considering temporal continuity of the scene, improving the segmentation accuracy of the railway environment.
2. The construction of a new dataset containing train front-view camera images and its annotations for the semantic segmentation of the railway environment.
3. The automatic construction of a class-labeled 3D map of the railway environment only from monocular camera images to improve the efficiency of the maintenance of railway facilities.

Through experimental analysis, we demonstrate the effectiveness of our proposed framework using a dataset of train front-view camera images and show an improvement on the semantic segmentation of the railway environment.

2 Semantic Segmentation Considering Temporal Continuity

2.1 Overview of the proposed method

As trains generally move in one direction, we can observe objects of the same class continuously in sequential frames of train front-view camera images. Ac-

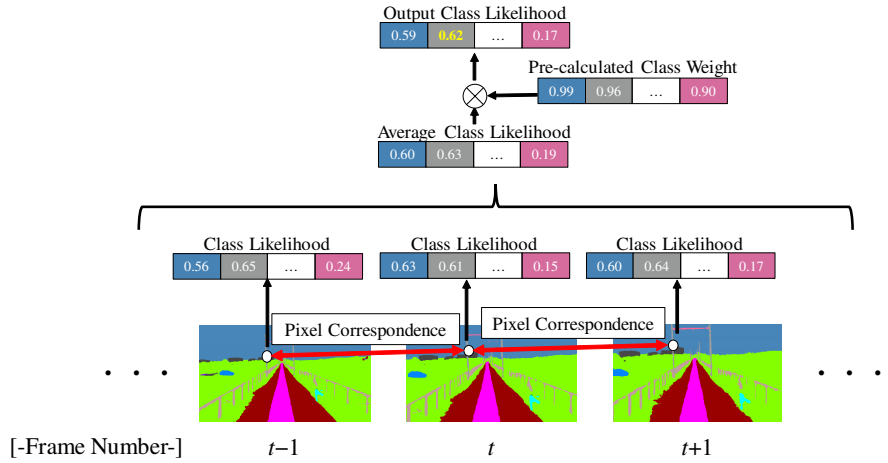

Fig. 2. Pixel-wise processing in the proposed method.

cordingly, we extend a conventional single-frame semantic segmentation method into a method that considers the temporal continuity of multiple frames to improve the accuracy of semantic segmentation.

In addition, there tends to be far less moving objects in train front-view camera images compared to those taken from vehicles, where there are numerous pedestrians and oncoming vehicles. Furthermore, when compared to vehicles that often make sharp turns, the direction of movement of trains change much more gradually. From such characteristics, we can infer that the optical flow between sequential frames of train front-view camera images can be obtained with high accuracy and density.

When trying to understand the railway environment, the recognition of small rail-side facilities are important. However, as such objects occupy comparatively small areas in an image, their semantic segmentation seems difficult compared to classes with larger areas like “sky” or “building”. To cope with this problem, we calculate each classes’ pixel occupancy ratio from training data containing annotated ground truths, and set a suitable weight for frame combination beforehand. Based on this weight, we can correct the class likelihood of a small rail-side facility that takes up a small area in an image to improve the detection rate of the corresponding class.

Fig. 1 shows the overall procedure of the proposed method. First, we apply conventional single-frame semantic segmentation to train front-view camera images and obtain likelihoods for each class per pixel. Then, we correspond each pixel in sequential frames, and by combining the class likelihoods of the corresponding pixels, output the multi-frame semantic segmentation results.

Fig. 2 shows an example of the pixel-wise procedure of the proposed method. First, the optical flow of sequential frames is calculated. Based on this, corre-

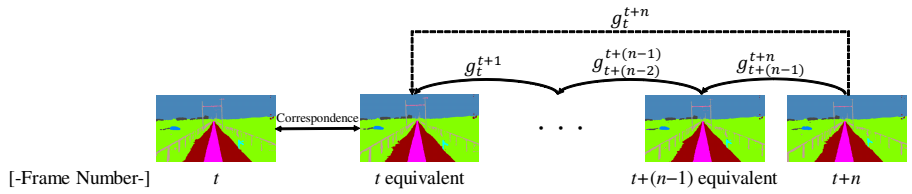

Fig. 3. Procedures for pixel correspondence between the focused frame and the n-th frame.

sponding pixels from sequential frames are estimated for each pixel within the focused frame by transforming the sequential frames. Next, we calculate the average class likelihood from single-frame semantic segmentation results of corresponding pixels in sequential frames. For each class value within the likelihood, we then multiply the pre-calculated weight for each class obtained from annotated training data. Finally, we select the class with the highest value within the weighted likelihood and output it as the result.

From here, we explain each step of the framework in detail.

2.2 Semantic segmentation using multiple frames

Correspondence of pixels using optical flow

In the proposed method, we correspond pixels between sequential frames by image transformation based on optical flow.

We consider a total of $2N + 1$ frames between the $(t - N)$-th frame to the $(t+N)$-th frame when focusing on the t-th frame. From here, when we state that we use "M" frames, M refers to the total number of frames used ($M = 2N+1$).

To start with, we first apply PWC-Net [10] to calculate the optical flow of adjacent frames. PWC-Net is a CNN that enables fast and accurate optical flow calculation with the use of pyramidal processing, warping, and cost volume. For frames taken prior to the focused t-th frame (past frames), we calculate the forward optical flow (f) between two sequential frame pairs. For frames taken after the focused frame (future frames), we similarly calculate the backward optical flow (g) of frame pairs. After applying PWC-Net, pixel-wise dense optical flow ($f_{t-(N-1)}^{t-N}, ..., f_t^{t-1}, g_t^{t+1}, ..., g_{t+(N-1)}^{t+N}$) is obtained.

Next, we calculate pixel correspondences based on the dense optical flow. Fig. 3 shows an overview of the process for finding pixel correspondence. To find out the correspondences of pixels to the pixel positions of the focused t-th frame, we transform past and future frames multiple times based on the calculated pixel movement vector (i.e. optical flow). For future frames, using the optical flow between the $(t + k)$-th frame and the $(t + k + 1)$-th frame g_{t+k}^{t+k+1}, we can transform the pixel position vector of the $(t + n)$-th frame $\boldsymbol{x}_{t+n}$ to $\hat{\boldsymbol{x}}_{t+n}$ using the equation below, and correspond it with the focused t-th frame.

Fig. 4. Example of missing pixels around image rims.

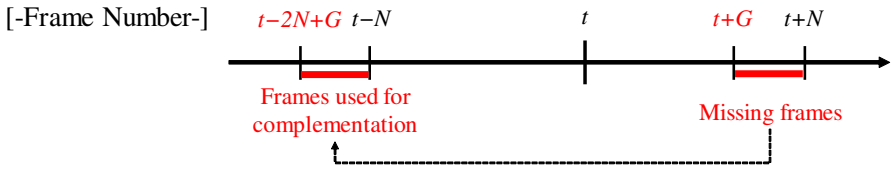

Fig. 5. Example of complementing the information for a missing pixel.

$$\hat{\boldsymbol{x}}_{t+n} = g_t^{t+n}(x_{t+n}) = g_t^{t+1} \circ g_{t+1}^{t+2} \circ \cdots \circ g_{t+n-2}^{t+n-1} \circ g_{t+n-1}^{t+n}(\boldsymbol{x}_{t+n}) \tag{1}$$

For past frames, similar procedures with the equation shown below can be used to convert the pixel position vector of the $(t-n)$-th frame $\boldsymbol{x}_{t-n}$ to $\hat{\boldsymbol{x}}_{t-n}$ to correspond it with the focused t-th frame.

$$\hat{\boldsymbol{x}}_{t-n} = f_t^{t-n}(x_{t-n}) = f_t^{t-1} \circ f_{t-1}^{t-2} \circ \cdots \circ f_{t-(n-2)}^{t-(n-1)} \circ f_{t-(n-1)}^{t-n}(\boldsymbol{x}_{t-n}) \tag{2}$$

Complementing the information of missing pixels around image rims When transforming images based on optical flow, some corresponding pixels exit outside future frame image rims where camera vision does not overlap. Fig. 4 shows examples of such cases. For such missing pixels, we complement the information with pixels from past frames to maintain the diversity of information and the denominator when calculating the average class likelihood. In particular,

when corresponding pixels in future frames are missing from the $(t+G+1)$-th frame, we refer to the pixel information from the $(t-N-1)$-th frame to the $(t-N-(N-G)=t-2N+G)$-th frame to maintain $2N+1$ frames worth of information. An example of this process is show in Fig. 5.

Decision of labels using pixel-wise class likelihoods

Now, we have the pixel position correspodence of multiple sequential frames with regards to the focused frame. Combining this with the semantic segmentation result of each frame, we calculate the pixel-wise class likelihood of the focused frame.

Firstly, we decide the class label c of each pixel within the focused frame from corresponding pixels' class likelihoods as follows.

$$c = \arg\max_{k \in K} \omega_k \bar{\ell}_k \tag{3}$$

The ω_k used here is a pre-set weight for each of the K classes. To be specific, because areas of rail-side facility classes within an image tend to be small, we pre-calculate the pixel sum ratio of each class labels within training data and decide the weight as follows.

$$\omega_k = 1 - \frac{\beta \alpha_i}{\sum_{k=0}^{K} \alpha_k} \tag{4}$$

Here, parameter α_i refers to the number of pixels that belong to class i in the training data, and β is a constant to adjust the sensitivity.

Meanwhile, $\bar{\ell}_k$, the class likelihood of class k, is calculated as the mean of class likelihoods of M number of frames. To be specific, it is calculated with the equation below referring to corresponding pixels from sequential frames.

$$\bar{\ell}_k = \frac{1}{M} \sum_{n=-N}^{+N} \ell_{t+n}^{k}(\hat{\boldsymbol{x}}_{t+n}) \tag{5}$$

Here, variable $\ell_{t+n}^{k}(\hat{\boldsymbol{x}}_{t+n})$ is the class likelihood of class k at pixel position vector $\hat{\boldsymbol{x}}_{t+n}$ at the $(t+n)$-th frame as calculated in Eq. (2). Thus, the output can be decided with the equation below.

$$c = \arg\max_{k \in K} \frac{\omega_k}{M} \sum_{n=-N}^{+N} \ell_{t+n}^{k}(\hat{\boldsymbol{x}}_{t+n}) \tag{6}$$

3 Experimental Evaluation

3.1 Class settings of the railway environment

In the proposed method, we calculate the class label of each pixel for unannotated train-front view camera images using a neural network trained with small hand-annotated samples of such images. For the neural network, DeepLabv3+ [2] is used. This is a neural network for semantic segmentation, with main features

Table 1. List of all semantic segmentation class settings.

Experimental class label	Corresponding Cityscapes [3] classes
flat	road, sidewalk, parking
building	building
construction	wall, guard rail, bridge, tunnel
fence	fence
pole	pole, pole group
traffic light	traffic light
traffic sign	traffic sign
nature	vegetation, terrain
sky	sky
human	person, rider
vehicle	car, truck, bus
train	train
two-wheel	motorcycle, bicycle
rail	—
track	—
level crossing	—
facility	—
crossing gate	—
overhead facility	—
railway light	—
railway sign	—

including atrous convolution and encoder-decoder structure. Table 1 shows the list of all semantic segmentation class settings used for this research. This is based on the class structure of the Cityscapes dataset [3], a dataset including semantic segmentation labels of vehicle front-view camera images, and some classes are combined or added to conform to the railway environment.

3.2 Datasets

In this research, we used images taken from a camera mounted in front of the driver's seat on a normally operated train. These images were taken by the Railway Technology Research Institute with the corporation of East Japan Railway Company. The train operated at a maximum of 85 km/h, which amounts to at most about 40 cm of forward movement as the video was taken at 60 frames per second. Furthermore, in addition to this train front-view camera image dataset, we used the Cityscapes dataset [3] for training neural networks for single-frame semantic segmentation.

For the training of DeepLabv3+ and the evaluation of the proposed method, we built a dataset from train front-view camera images. First, we manually selected twelve frames from such images ($1,920 \times 1,080$ pixels) so that they contain a variety of different objects like crossings and signs. We then annotated

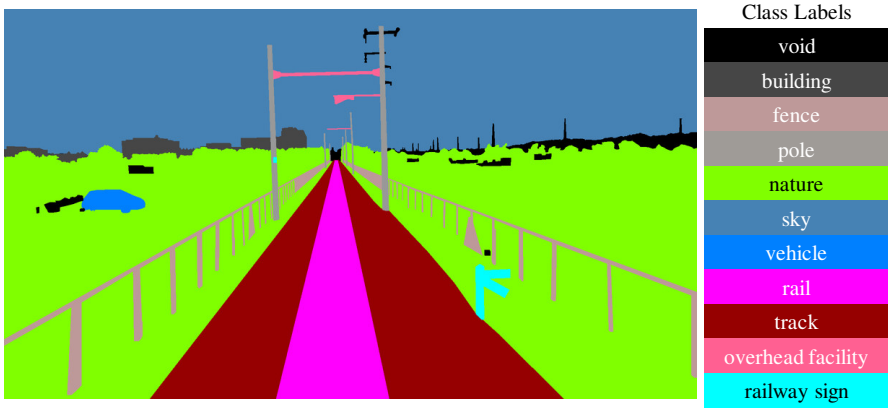

Fig. 6. Annotation example of the train front-view camera image dataset.

semantic segmentation labels for each pixel in each frame by hand. An example of the annotation can be seen in Fig. 8.

3.3 Experiment

We conducted an experiment to evaluate the effectiveness of considering temporal continuity on semantic segmentation of train front-view images. We tested the following four methods:

City-1 Conventional single-frame semantic segmentation using Deeplabv3+, trained using only the Cityscapes dataset.

Rail-1 Conventional single-frame semantic segmentation using Deeplabv3+, pre-trained using the Cityscapes dataset and fine tuned using the train front-view camera image dataset.

Label Majority Multi-frame semantic segmentation that uses sequential frames' pixel correspondence and decides the output label as the majority of the corresponding pixels' class labels.

Weighted Mean Proposed multi-frame semantic segmentation that uses sequential frames' pixel correspondence and decides the output label according to the weighted mean of the corresponding pixels' class likelihoods.

On conducting the experiment, we set the number of frames used to 3, 5, or 7. Since the train front-view camera image dataset consists of only twelve images, we use transfer learning. After pre-training Deeplabv3+ with the Cityscapes dataset, we only initialize the weights between the last and the second to the last layer of the network, and re-train the network using the train front-view camera dataset. We also split the dataset into four sections, and apply cross validation to calculate the mean of the four results. When splitting the dataset, we try to contain each target class evenly within each section.

Table 2. Class IoU and mIoU for each method.

Method	City-1	Rail-1	Label majority			Weighted mean		
Frames used (M)	1	1	3	5	7	3	5	7
flat	*0.0195*	0.0259	0.0204	0.0244	0.0243	0.0363	**0.0776**	0.0578
building	*0.3482*	0.5684	0.5721	**0.5918**	0.5913	*0.5684*	0.5827	0.5845
construction	*0.0899*	0.1653	0.1690	0.1757	0.1807	0.1804	0.1844	**0.2181**
fence	*0.1153*	0.4639	0.4643	0.4590	0.4267	0.4829	**0.4896**	0.4705
pole	*0.3501*	0.4927	0.4905	0.4880	0.4541	**0.5007**	0.4741	0.4120
traffic light	—	—	—	—	—	—	—	—
traffic sign	—	—	—	—	—	—	—	—
nature	*0.5489*	0.7511	0.7531	0.7553	0.7516	0.7565	**0.7597**	0.7576
sky	*0.8945*	0.9258	0.9240	0.9233	0.9211	**0.9269**	0.9262	0.9229
human	—	—	—	—	—	—	—	—
vehicle	0.5246	0.5312	0.5306	0.5294	0.5298	**0.5334**	0.5292	*0.5244*
train	—	—	—	—	—	—	—	—
two-wheel	—	—	—	—	—	—	—	—
rail	*0.0000*	0.8837	0.8854	0.8770	0.8696	0.8869	**0.8887**	0.8853
track	*0.0000*	0.8283	0.8280	0.8253	0.8206	0.8312	0.8365	**0.8376**
level crossing	0.0000	0.0000	**0.0011**	0.0000	0.0000	0.0000	0.0000	0.0000
facility	*0.0000*	0.2185	0.2167	0.2363	0.2326	0.2316	**0.2406**	0.2164
crossing gate	*0.0000*	**0.1449**	0.0721	0.0489	0.0211	0.1305	0.0449	0.0153
overhead facility	*0.0000*	0.3115	0.2977	0.2830	0.2721	**0.3264**	0.3218	0.2894
railway light	*0.0000*	0.4646	0.4612	0.4808	**0.5123**	0.4848	0.5017	0.4936
railway sign	*0.0000*	0.2784	0.2780	0.2227	0.2761	**0.2855**	0.2431	0.2511
mIoU	*0.2041*	0.4917	0.4909	0.4898	0.4795	**0.4977**	0.4930	0.4785

For evaluation, we use mean intersection over union (mIoU) and class intersection over union (class IoU) as metrics. These are calculated using the area ρ_k of a given class k within an image. As our dataset is small in size, after splitting it into four, we may have cases where some classes only appear in the training data, and not appear in the testing data. To cope with such cases, we only calculate the class IoU of the classes that appear in both the training and the target data, and calculate mean IoU as the mean of all such class IoUs.

Table 2 shows the class IoU and the mIoU of the semantic segmentation results for each method. As the result of semantic segmentation considering temporal continuity, we can see that from the baseline single-frame method trained using only the Cityscapes dataset, the mIoU improved by about 28.7% when combining it with the train front-view camera image dataset. The mIoU further improved by about 0.6% in the proposed method which uses $M = 3$ frames and their class likelihoods. Note that in Table 2, classes without results did not appear in any of the ground-truth data for this experiment.

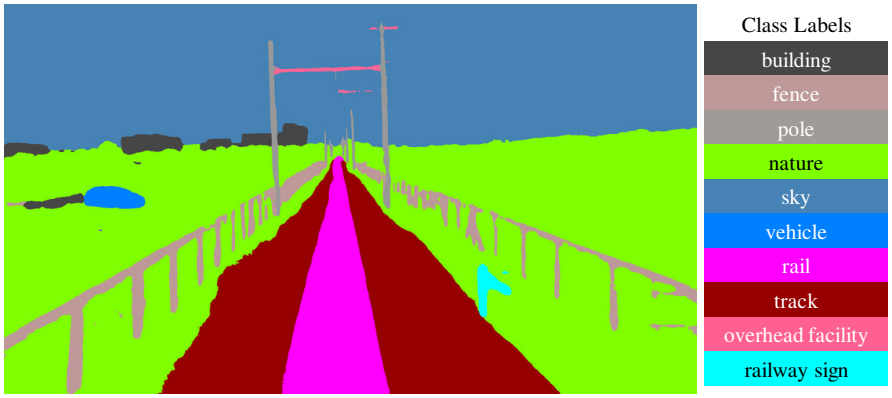

Fig. 7. Output examples of semantic segmentation by the proposed method.

4 Discussion and Applications

4.1 Improvement of semantic segmentation accuracy

First, we look at the experimental results shown in Table 2. The mIoU for the semantic segmentation improved from 20.4% in the baseline single-frame method (City-1) to 49.17% in the modified single-frame method (Rail-1); An improvement of 28.7%. This is simply due to the availability of an appropriate training data, since the Cityscapes dataset does not include any image of the railway environment. The mIoU also improved from the modified single-frame method to the proposed method 3 (Weighted Mean) with $M = 3$, by 0.6%, and by 0.1% with $M = 5$ frames. Such results may come from cases in class borders where the conventional single-frame semantic segmentation outputs incorrect labels, but using multiple frames considering temporal continuity helped stabilize the output. However, when $M = 7$, the mIoU actually decreased, which could suggest that simply increasing the number of frames does not correlate to improving the overall accuracy.

When looking at each class IoU result, we can see that the proposed method with $M = 3$, the results improved for all classes that were added for the railway environment other than "crossing gate". As stated before, we weighted classes that take up small areas in training images. This proved to be effective as class likelihoods of classes like "rail" and "track" are boosted. Also, for classes like "nature" and "sky" that take up large areas, there was no significant decrease in the class IoUs, suggesting that the class weights used in the proposed method effectively adjusted the class likelihoods of classes on small objects, while maintaining the results for those on large objects.

4.2 Effects of using class likelihoods

The proposed method outperformed a similar method that outputs the majority of class labels, especially in cases where the number of frames used was small.

$(N+1)$-th frame $(N+3)$-th frame

Fig. 8. Visulaized flow estimation results corresponding to the N-th frame. Each frame was warped to match the N-th frame using the estimated flow. Red and blue areas indicate pixels where the N-th frame and the warped image differ.

This result may have been influenced by sampling that occurs when images were taken. In particular, boundaries between two classes can become vague due to such sampling. In vague boundaries, both class likelihoods become small. However, by combining pixel information of multiple frames, we can use class likelihoods of corresponding pixels where class boundaries are clearer to estimate the semantic label. In such cases, the proposed method could comprehensively compare all class likelihoods of corresponding pixels and decide an appropriate output class label. Meanwhile, with the use of the majority of class labels, class labels are decided per frame even in vague class boundaries, which lead to a decrease in precision in such cases.

4.3 Accuracy of flow estimation

The proposed method used PWC-Net to estimate the flow between multiple frames, and used its output to correspond pixels among the frames. If the pixels were corresponded perfectly, the result scores of the proposed method should at least remain the same when more frames are used. However, the result in Table 2 shows that as more frames were used, the overall mIoU decreased. This result may be due to the limited accuracy of flow estimation. An example of flow estimation is visulalized in Fig. 1. The estimated flow of the $(N+3)$-th frame has more difference to the original frame compared to the flow of the $(N+1)$-th frame, especially around edges of vertical objects like fences. As the number of used frames increases, flow estimation error will also accumulate, resulting in degraded overall performance of the proposed method. To alliviate the effects of flow estimation on overall IoU, better optical flow estimation methods that consider the characteristics of the railway environment are required.

4.4 Possible applications to the railway environment

With the proposed method, we can obtain pixel-wise label information of images of the railway environment. For the purpose of railway environment recognition,

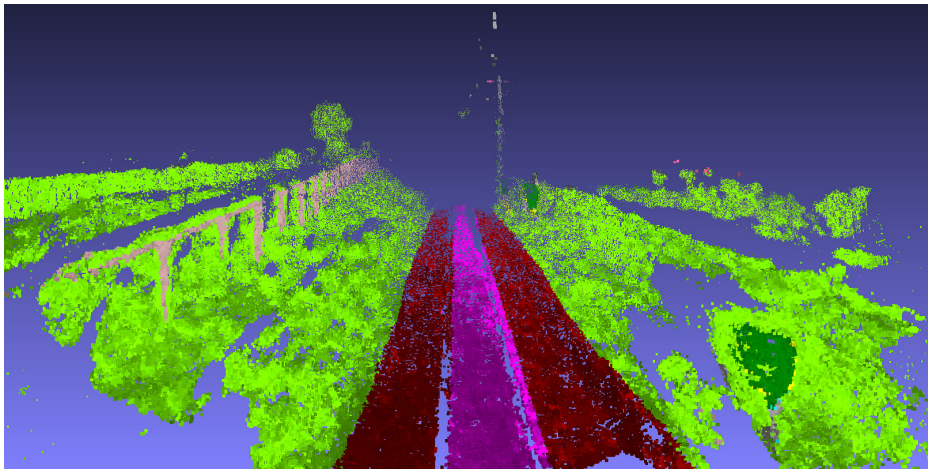

Fig. 9. Combining SfM with semantic segmentation.

we can apply methods like Structure from Motion (SfM) [9] to reconstruct 3D point clouds from a series of images. Combining the semantic segmentation results of 2D images and such 3D reconstruction enables us to build a class labeled 3D map of the railway environment. An example of such application is shown in Fig. 9. The pink points represent the "rail" class, and the red ones represent the "track" class. Other classes like "pole"(gray), "facility"(dark green), "fence"(light orange), and "nature"(light green) can also be seen. However, the accuracies of both semantic segmentation and SfM reconstruction are still insufficient for practical use. For the semantic segmentation side, using far more training data would likely improve the result. Meanwhile, SfM reconstruction of the railway environment is difficult as a monocular camera can only move forward, and the perspective of the images do not change dramatically. One idea to improve the SfM reconstruction would be to use extra information like semantic labels to post-process the 3D point clouds. This may give us more accurate labeled 3D point clouds, which we can then use for the maintenance of rail-side facilities in real-world environments.

5 Conclusion

In this paper, we proposed a method that improves the accuracy of semantic segmentation on train front-view camera images with the use of multiple frames and their optical flow to consider temporal continuity. Assuming that the lateral movement of trains are smooth, we used information from multiple frames to consider temporal continuity during semantic segmentation. We also constructed a new dataset consisting of train front-view camera images and its annotations for semantic segmentation.

Experimental results show that the proposed method outperforms the conventional single-frame semantic segmentation, as well as the effectiveness of the use of class likelihoods over class labels.

Future works include the mutual improvement of both semantic segmentation and SfM reconstruction accuracy, as well as experimenting the proposed method using a larger train front-view camera dataset.

Acknowledgement

Parts of this research were supported by MEXT, Grant-in-Aid for Scientific Research.

References

1. Badrinarayanan, V., Kendall, A., Cipolla, R.: SegNet: A deep convolutional encoder-decoder architecture for image segmentation. IEEE Trans. on Pattern Analysis and Machine Intelligence **39**(12), 2481–2495 (12 2017)
2. Chen, L.C., Zhu, Y., Papandreou, G., Schroff, F., Adam, H.: Encoder-decoder with atrous separable convolution for semantic image segmentation. In: Proc. 15th European Conf. on Computer Vision. pp. 801–818 (9 2018)
3. Cordts, M., Omran, M., Ramos, S., Rehfeld, T., Enzweiler, M., Benenson, R., Franke, U., Roth, S., Schiele, B.: The Cityscapes dataset for semantic urban scene understanding. In: Proc. 2016 IEEE Conf. on Computer Vision and Pattern Recognition. pp. 3213–3223 (6 2016)
4. Gibert, X., Patel, V.M., Chellappa, R.: Material classification and semantic segmentation of railway track images with deep convolutional neural networks. In: Proc. 2015 IEEE Int. Conf. of Image Processing. pp. 621–625 (9 2015)
5. Kundu, A., Li, Y., Dellaert, F., Li, F., Rehg, J.M.: Joint semantic segmentation and 3d reconstruction from monocular video. In: Proc. 13th European Conf. on Computer Vision. pp. 703–718. Springer (9 2014)
6. Long, J., Shelhamer, E., Darrell, T.: Fully convolutional networks for semantic segmentation. In: Proc. 2015 IEEE Conf. on Computer Vision and Pattern Recognition. pp. 3431–3440 (6 2015)
7. Niina, Y., Oketani, E., Yokouchi, H., Honma, R., Tsuji, K., Kondo, K.: Monitoring of railway structures by mms. Journal of the Japan society of photogrammetry and remote sensing **55**(2), 95–99 (2016). https://doi.org/10.4287/jsprs.55.95
8. Ronneberger, O., Fischer, P., Brox, T.: U-net: Convolutional networks for biomedical image segmentation. In: Proc. 2015 Int. Conf. on Medical Image Computing and Computer-Assisted Intervention. pp. 234–241 (11 2015)
9. Schonberger, J.L., Frahm, J.M.: Structure-from-motion revisited. In: Proc. 2016 IEEE Conf. on Computer Vision and Pattern Recognition. pp. 4104–4113 (6 2016)
10. Sun, D., Yang, X., Liu, M.Y., Kautz, J.: PWC-Net: CNNs for optical flow using pyramid, warping, and cost volume. In: Proc. 2018 IEEE Conf. on Computer Vision and Pattern Recognition. pp. 8934–8943 (6 2018)